Towards a Hierarchy of Cryptographic Protocol Models

Catherine Meadows
Center for High Assurance Computer Systems
Naval Research Laboratory
Code 5543
Washington, DC 20375
meadows@itd.nrl.navy.mil

ABSTRACT

Recently there has been an increasing amount of research in the introduction of cryptographic ideas into discrete methods for cryptographic protocol analysis. This is often done by developing a discrete model and a cryptographic model such that the discrete model can be shown sound with respect to the cryptographic model. In this position paper we talk about some of the other issues in cryptographic protocol analysis that could be addressed with this approach, and propose a hierarchy of models.

Categories and Subject Descriptors

C.2.2 [**Network Protocols**]: security, protocol verification, theory

General Terms

cryptographic protocols, formal methods

1. INTRODUCTION

The application of formal methods to the analysis of cryptographic protocols has been an active field of research in the past few years. However, the approach of using discrete methods to analyze systems that are based on cryptographic algorithms whose security is based on probabilistic and complexity theoretic arguments has raised some questions. Although the use of this approach has led to the successful discovery of bugs, any security proof it provides rests on assumptions that are for the most part unverified. The realization of this lack has led to the recent research in the development of security models amenable to discrete analysis that can be proved sound with respect to more detailed cryptographic models. Such results can be helpful even when techniques already exist for proving correctness with respect to a cryptographic model by hand; the fact that the formal techniques can be automated allows one to be sure that the proof is error-free and that all conditions have been covered.

FMSE'03, October 30, 2003, Washington, DC, USA.

The point of this position paper is that the approach of using a simpler and more tractable model that can be proven sound with respect to a more complex and detailed model is useful for handling other assumptions than cryptographic ones. Ultimately, it may make more sense to have a hierarchy of models rather than a two-tiered system. We develop these ideas in more details below.

2. SOME ISSUES

In this section we discuss some of the issues that we believe could be addressed by developing appropriate abstractions in a hierarchy of models.

2.1 Algebraic Properties of Operations

When modeling a cryptographic protocol, one assumes that it obeys a certain set of algebraic identities. For example, one might want to assume that encryption and decryption cancel each other out, or that exponentiation for Diffie-Hellman commutes. For the sake of an efficient analysis, one would like to use as few of these identities as possible. Thus, for example, most formal models for cryptographic protocol analysis model the system as a free algebra where only the encryption operator is modeled explicitly. The decryption operator is modeled implicitly in the protocol rules, and by stating that if an attacker knows an encrypted term and the key it was encrypted with, then it can produce the term. Although there are examples of protocols that can not be modeled or analyzed using the free algebra model (see for example the Simmons Selective Broadcast protocol analyzed by us in [3]), it appears to be adequate for most protocols. However, it was not clear in which cases the free algebra model was sound with respect to the more detailed model, until Millen recently described a class of protocols for which this soundness result holds [5].

There are other algebraic properties for which it would be helpful to have similar results. What properties are necessary to model Diffie-Hellman, and under what circumstances? For example, such questions arise about systems such as the protocols developed by Ateniese et al. in [1] which make use of properties of Diffie-Hellman exponentiation such as commutativity and the existence of inverses. The homomorphic properties of RSA are necessary to model primitives such as blinded signatures. Can these properties ever be abstracted away from, and how? Can we safely abstract away from other properties of RSA? What about the algebraic properties of exclusive-or?

2.2 Representing System Failures

A protocol should be designed to be secure against certain kinds of system failures such as, for example, compromise of old session keys. Although some attention has been paid to this in the formal analysis of cryptographic protocols, many formal systems ignore this problem, and others model only some types of vulnerabilities (compromise of session keys, but not master keys, for instance). On the other hand, many formulations of cryptographic soundness have well-defined failure models. Can these be used to enrich the failure models of the formal systems?

2.3 Disambiguating Messages

There are a number of attacks on cryptographic protocols that rely on an attacker's ability to pass off a message of one kind as a message of another. For example, we found such an attack in our analysis of an early version of the Group Domain of Interpretation (GDOI) protocol [4]. Heather et al. [2] have shown that it is possible in certain circumstances to guarantee security against typing attacks if unambiguous formatting is used. What are the best ways of identifying and removing vulnerabilities that arise from ambiguous formatting? Can this be done at the start of the security analysis, so that during the rest of the analysis one can rely on the protocol's being free from typing attacks? When can we enforce use of an unambiguous formatting system, and how can we deal with possible interactions with legacy systems which may not use unambiguous formatting?

3. THE HIERARCHY AND ITS USES

Once we have identified the different issues that need to be addressed, we can develop a hierarchy of models at varying degrees of abstraction. At the top of the hierarchy we would have the classic model used by most designers of analysis tools for cryptographic protocols: cryptosystems modeled by free algebras, strong typing, and the only failure mode being that of a node under the complete control of an adversary. At a slightly lower level we would model different algebraic properties of the various operations used (we would probably have several levels of algebraic properties, as a matter of fact). At other levels we would have different types of system failures. At still other levels we would model formatting conventions. At the bottom level, or near to the bottom, we would introduce the cryptographic models that underly the protocols.

When we wished to analyze a protocol, we would pick the maximal model such that an analysis of that protocol with respect to that model could be proved sound with respect to all the models below it. The exact model that we could choose would depend our ability to prove that the protocol satisfies certain conditions. Of course, it would be in our interest to prove that a protocol satisfies the conditions that allow us to use the maximal model, but for practical reasons we might not be always be able to do this. In that case, we would choose the highest ranking model feasible.

4. CONCLUSION

In this position paper we have given a brief outline of a strategy for rendering the analysis of cryptographic protocols by formal methods both sound and tractable. We believe that the incorporation of cryptographic models is an important part of this strategy, but this is only part of the picture. We have given a brief overview of some of the other issues that need to be considered, and we believe that others will be identified in the future.

5. REFERENCES

[1] G. Ateniese, M. Steiner, and G. Tsudik. New multiparty authentication services and agreement protocols. *IEEE Journal on Selected Areas in Communications*, 18(4), April 2000.

[2] J. Heather, S. Schneider, and G. Lowe. How to prevent type flaw attacks on security protocols. In *Proceedings of the 13th IEEE Computer Security Foundations Workshop*. IEEE Computer Society Press.

[3] C. Meadows. Applying formal methods to the analysis of a key management protocol. *Journal of Computer Security*, 1(1), January 1992.

[4] C. Meadows, P. Syverson, and I. Cervesato. Formal specification and analysis of the GDOI protocol using NPATRL and the NRL protocol analyzer. *Journal of Computer Security*, 2003. to appear.

[5] J. Millen. On the freedom of decryption. *Information Processing Letters*, 86(6):329–323, June 2003.